# APS105 Lecture 17
# 7.3

Dan Zingaro

March 5, 2010

# Feedback from Reading Quiz

- "Do I have to memorize the names of the string functions?"
  - Yes, please
- "What does `strstr` do when there are multiple occurrences of `s2`?"
  - It returns a pointer to the first occurrence

# String Comparisons

```
int strcmp (const char *s1, const char *s2)
```

- Compares the two strings based on dictionary order
- Returns a negative number if `s1` goes before `s2` in dictionary order
- returns `0` if the strings are equal
- Returns a positive number if `s1` goes after `s2` in dictionary order
- e.g. `strcmp ("bar", "car")` returns a negative number
- e.g. `strcmp ("cat", "catch")` returns a negative number
- e.g. `strcmp ("catch", "cat")` returns a positive number

# String Comparisons...

- `strcmp` examines the strings character by character, until we hit a `'\0'` or we find a mismatch
- The strings are equal if we get to the end of both strings
  - e.g. `cat` and `cat`
- One string is less than another if it ends first
  - e.g. `cat` and `catch`
- If we find a character mismatch, the string with the smaller character is less than the other
  - e.g. `cards` and `cord`

# ConcepTest

Here is an implementation of `strcmp`, with the while-condition removed. Which while-condition makes the code correct?

```
int my_strcmp (const char *s1, const char *s2) {
  while (...) {
    s1++;
    s2++;
  }
  if (*s1 == '\0' && *s2 == '\0') return 0;
  else if (*s1 == '\0') return -1;
  else if (*s2 == '\0') return 1;
  else return *s1 - *s2;
}
```

- A. `*s1 == *s2`
- B. `*s1 == *s2 && *s1 != '\0' && *s2 != '\0'`
- C. `*s1 == *s2 || *s1 != '\0' || *s2 != '\0'`
- D. A and B
- E. B and C

# String Copying

- We cannot copy a string by using an assignment

```
char s1[5] = "hi";
char s2[5] = "bye";
s1 = s2; //Wrong!
```

- Instead, we use the `strcpy` string function

```
char *strcpy (char *s1, const char *s2)
```

- Copies the characters from `s2` (including `'\0'`) into `s1`
- Make sure `s1` has enough space for `s2`'s characters!

# String Concatenating

```
char *strcat (char *s1, const char *s2)
```

- Appends the characters of `s2` (including `'\0'`) to those of `s1`
- Make sure `s1` has enough space for the characters it already has, plus those of `s2`!

# ConcepTest

What is stored in `s` after the following code executes?

```
char s[10] = "one";
char t[10] = "two";
strcpy (s, t);
strcpy (t, s);
strcat (s, t);
```

- A. `one one`
- B. `two two`
- C. `one two two one two`
- D. `two`
- E. `one`

# Searching for a Character

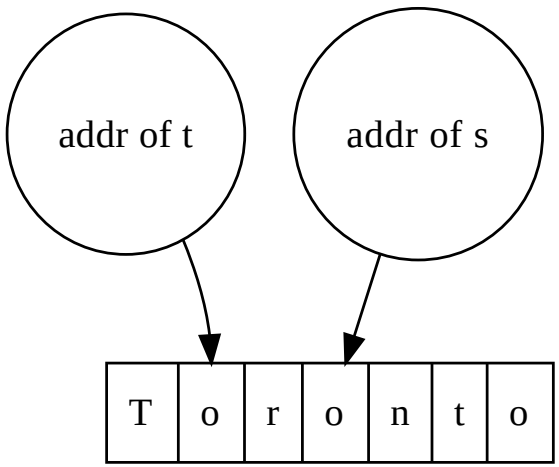
```
char *strchr (const char *s, int c)
```

- Returns a pointer to the first occurrence of character `c` in string `s`
- If `c` does not occur at all in `s`, it returns a null pointer
- A null pointer is a pointer that points nowhere; we cannot use `*` on it to get the object it points to
- We can test for a null pointer

```
if (p == NULL)
  ...
```

# Searching for a Character...

```
char s[] = "toronto";
char *t;
t = strchr (s, 'o');
```




- Now, `t` points to the first `o` in `Toronto`
- `t` is not an integer
- `*t == 'o'`
- `*(t + 1) == 'r'`

# Searching for a Substring

```
char *strstr (const char *s1, const char *s2)
```

- Returns a pointer to the first character of the first occurrence of string `s2` in string `s1`
- If `s2` does not occur at all in `s1`, it returns a null pointer

# ConcepTest

Assume string `s2` exists in string `s1`. We want to know the index of `s1` at which `s2` starts. For example, if `s1` is `school` and `s2` is `ool`, we want the value 3. Which of the following lines of code does this?

- A. `strstr (s1, s2) - s2`
- B. `strstr (s1, s2) - s1`
- C. `strstr (s1, s2) - s2 + 1`
- D. `strstr (s1, s2) - s1 + 1`